

LẬP TRÌNH MẠNG CĂN BẢN

Biên soạn: ThS. Đỗ Thị Hương Lan



TRƯỜNG ĐH CÔNG NGHỆ THÔNG TIN - ĐHQG-HCM
KHOA MẠNG MÁY TÍNH & TRUYỀN THÔNG
FACULTY OF COMPUTER NETWORK AND COMMUNICATIONS

Tầng 8 - Tòa nhà E, trường ĐH Công nghệ Thông tin, ĐHQG-HCM

Chương 7

Lập trình với các giao thức làm việc với Email

Nội dung

**Giao thức SMTP, POP3, IMAP
và Ứng dụng gửi – nhận Email**

Nội dung chi tiết

- Giới thiệu về email và nguyên tắc hoạt động
- Giao thức SMTP
- Giao thức POP3 và IMAP
- Lập trình Gửi Mail
- Lập trình Nhận Mail

Nội dung chi tiết

- Giới thiệu về email và nguyên tắc hoạt động
- Giao thức SMTP
- Giao thức POP3 và IMAP
- Lập trình Gửi Mail
- Lập trình Nhận Mail

Giới thiệu

- Email là thư điện tử được sử dụng trong môi trường Internet để người dùng có thể trao đổi thông điệp với nhau.
- Email là một cách thức giao tiếp quan trọng và phổ biến nhất trong mọi lĩnh vực đời sống, xã hội ...
- Lập trình quản lý email là vấn đề quan trọng và cần thiết

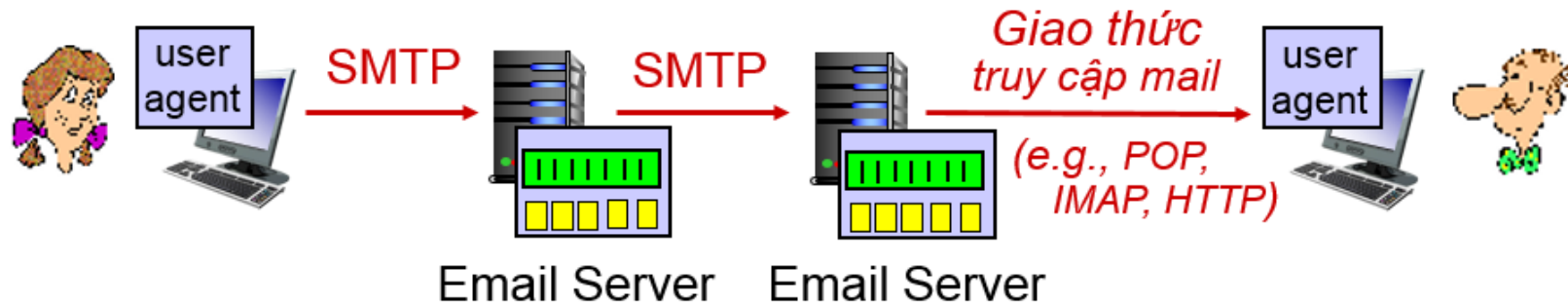


Email là gì?

- Mỗi email có địa chỉ duy nhất, dạng: **<Username>@<domain name>**
- Với:
 - <domain name> duy nhất trên hệ thống DNS toàn cầu
 - <Username> chỉ cần duy nhất trong mail server của người nhận

Ví dụ: 16520010@gm.uit.edu.vn, tranvana@gmail.com

Nguyên tắc hoạt động của việc gửi nhận mail



- **Gửi email:** Sử dụng SMTP
- **Nhận email:** Có 2 giao thức chính là POP (110) và IMAP (143)
- **Webmail:** Sử dụng giao thức HTTP (dựa trên Webmail) để gửi và nhận email.

Gửi email

- Email không được gửi trực tiếp từ người gửi đến người nhận mà **thông qua các Mail Server**.
- Khi người gửi soạn và gửi đi một email, thư này sẽ được xếp tại **hàng chờ** (Mail Queue) của Mail Server (người gửi).
- Mail Server của người gửi sẽ chuyển email này đến Mail Server của người nhận.

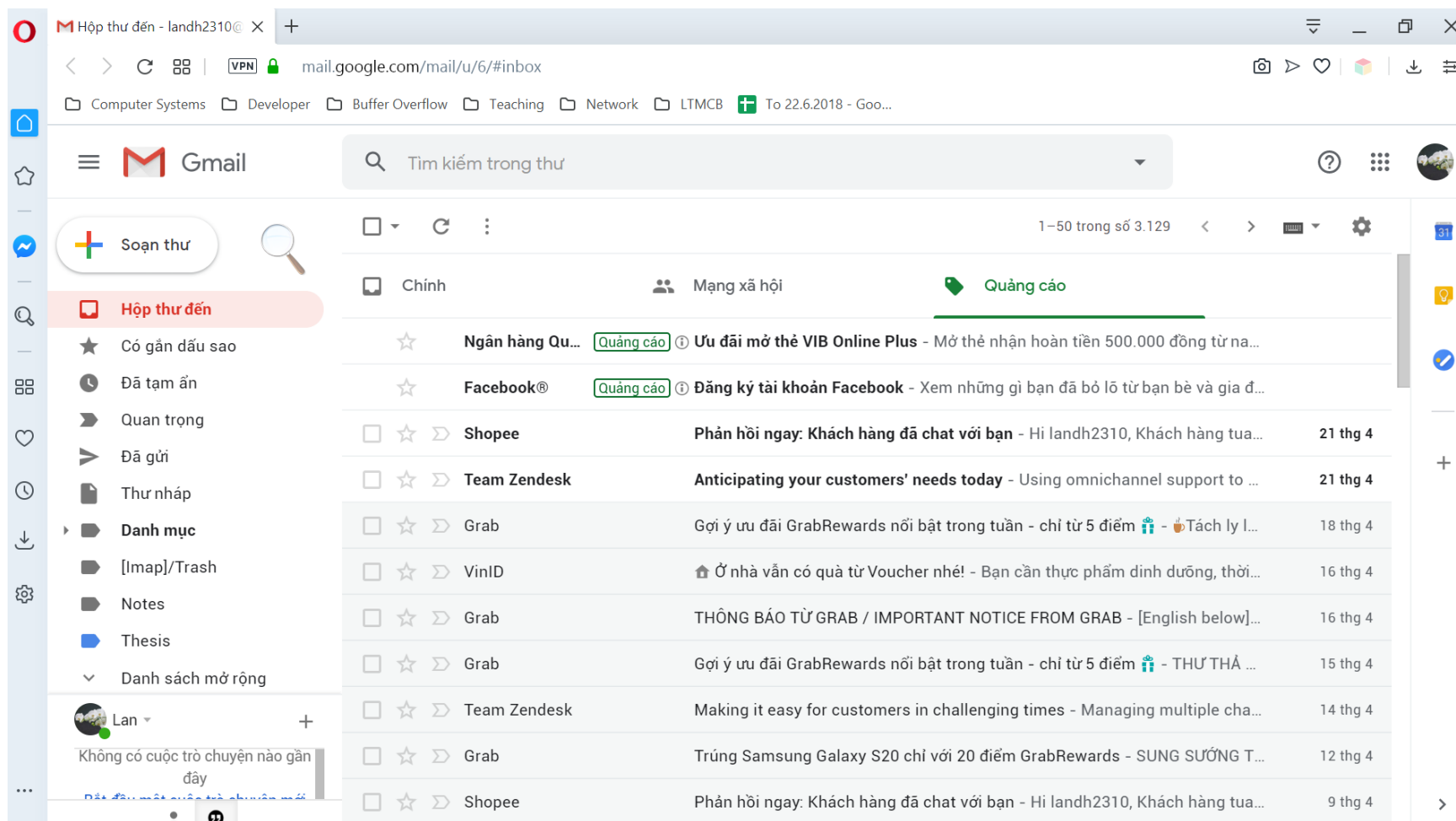
Nhận email/Truy xuất email

- Email được lưu trữ tại hộp thư của người nhận
- Sử dụng các giao thức truy xuất mail
 - **Kết nối** đến Mail Server
 - **Lấy** nội dung email

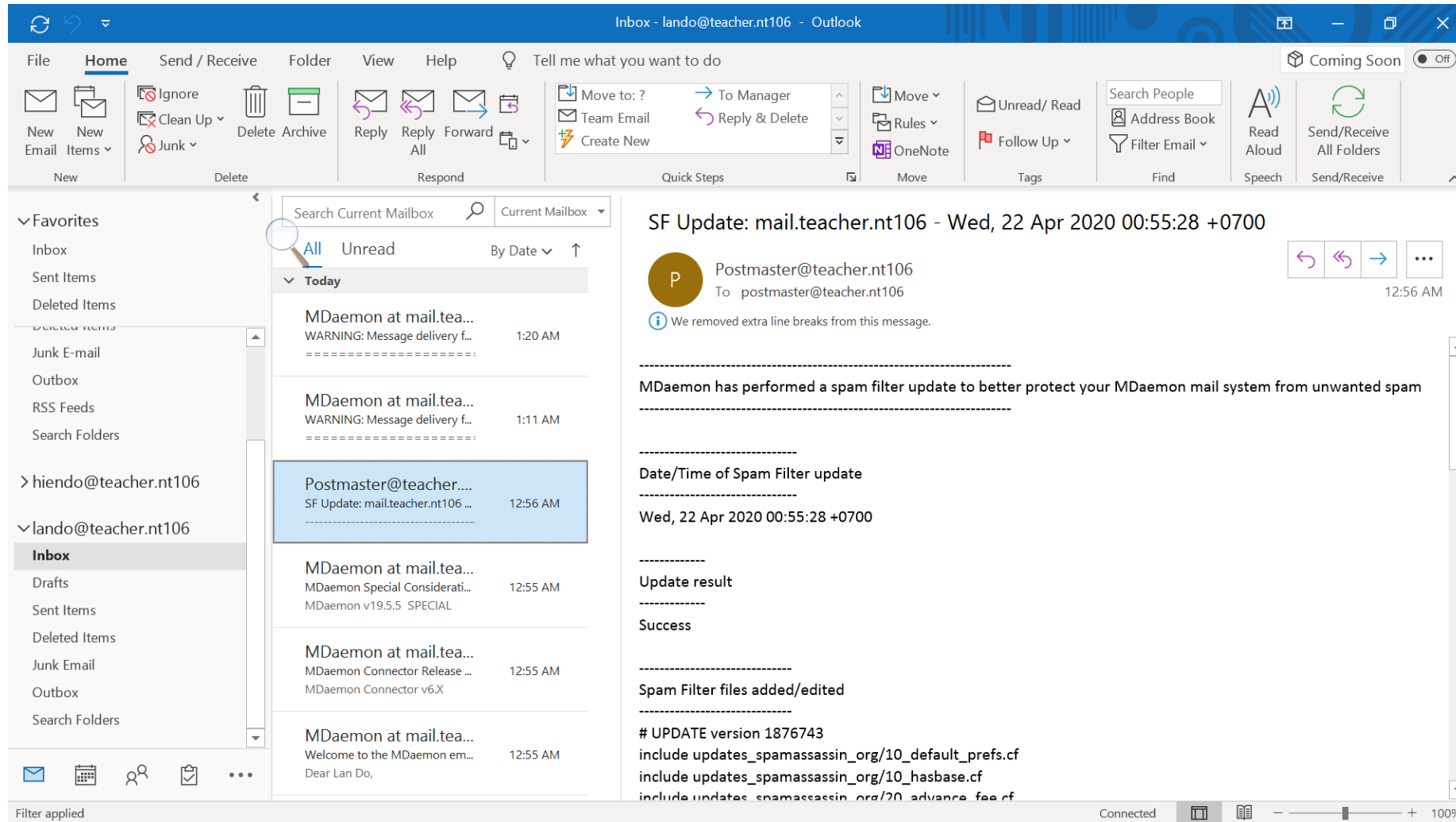
Email clients và Webmail

- Webmail: truy cập email bằng trình duyệt
- Email client: ứng dụng quản lý email
- Cả 2 đều là ứng dụng **gửi và nhận** email

Webmail



Email Client



Mail Server



Nội dung chi tiết

- Giới thiệu về email và nguyên tắc hoạt động
- **Giao thức SMTP**
- Giao thức POP3 và IMAP
- Lập trình Gửi Mail
- Lập trình Nhận Mail

Giới thiệu

- Email là thư điện tử được sử dụng trong môi trường Internet để người dùng có thể trao đổi thông điệp với nhau.
- Email là một cách thức giao tiếp quan trọng và phổ biến nhất trong mọi lĩnh vực đời sống, xã hội ...
- Lập trình quản lý email là vấn đề quan trọng và cần thiết



SMTP – Simple Mail Transfer Protocol

- Giao thức **chỉ dùng để gửi email**
- Mọi mail server phải tuân theo chuẩn SMTP để gửi được đến đích (RFC 821)
- Cài đặt một mail server đơn giản:
 - Hướng dẫn trong bộ cài đặt Windows Server
 - Dùng phần mềm của bên thứ 3: AspEmail, Mercury, Pegasus, MDaemon ...

Gửi SMTP mail

- Dùng TCP port 25
- Dùng telnet để kiểm tra địa chỉ IP của ISP
 - Mở cmd
 - Nhập lệnh: **telnet <server name> 25**
 - Khi client đã kết nối được với server thì server luôn trả lời với nội dung:
220 <lời chúc mừng> <version-number>

PHIÊN LÀM VIỆC VỚI SMTP

- Một phiên làm việc:

HELO/EHLO

MAIL FROM: <*địa chỉ email của người gửi*>

RCPT TO: <*địa chỉ email của người nhận*>

DATA

<*nội dung của email*>

.

QUIT

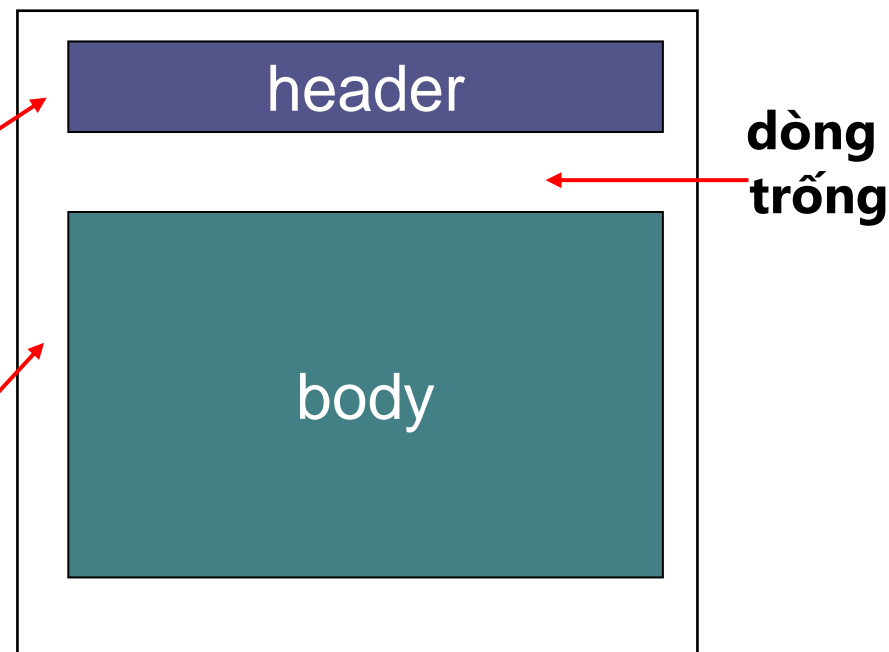
Định dạng thông điệp Mail

RFC 822: chuẩn cho định dạng thông điệp văn bản:

- Các dòng header, ví dụ
 - To:
 - From:
 - Subject:

Khác với các lệnh SMTP MAIL
FROM, RCPT TO!

- Body: “thông điệp”
 - Chỉ có các ký tự ASCII

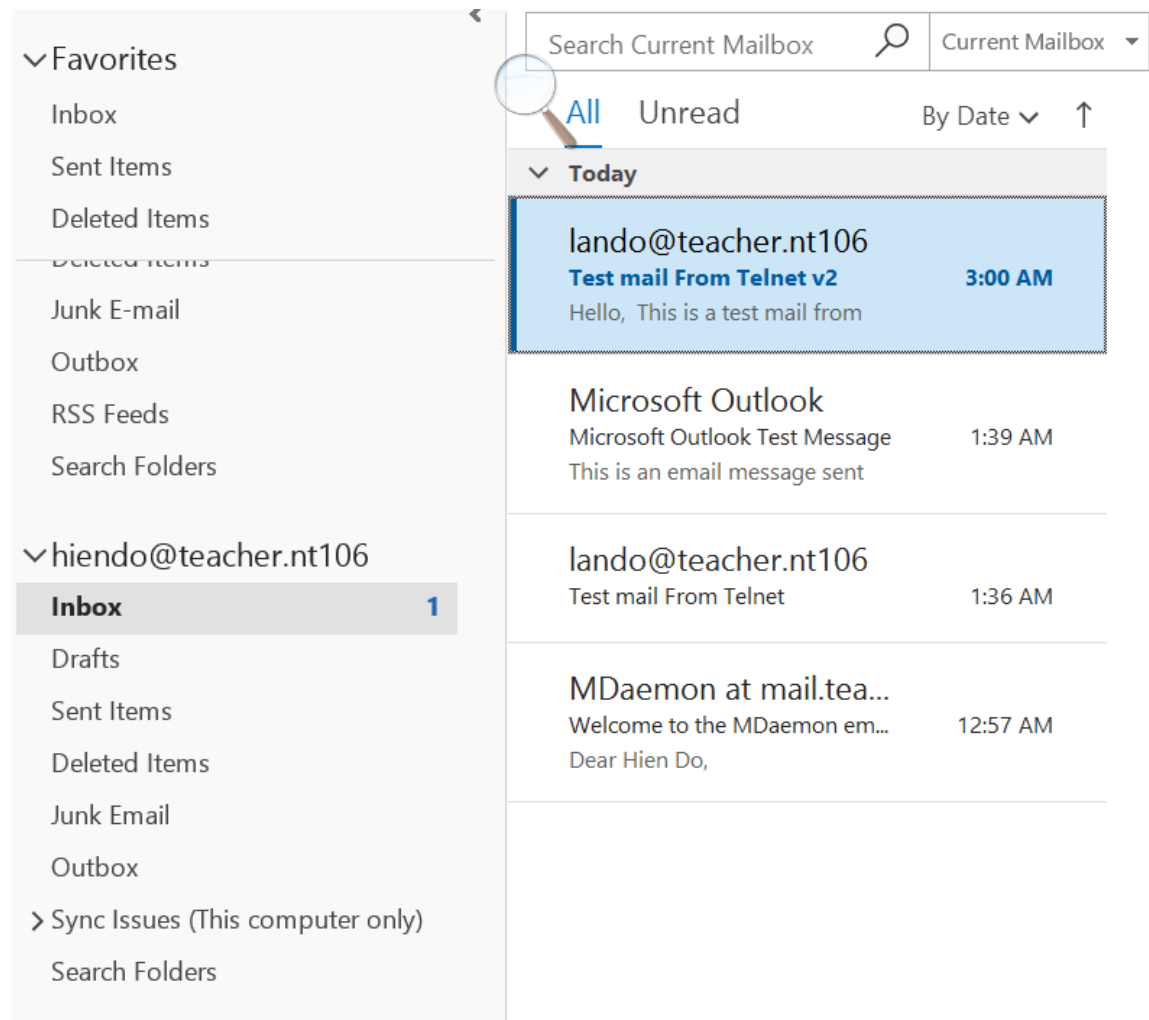


Gửi mail với Telnet

```
C:\Windows\system32\cmd.exe
220 mail.teacher.nt106 ESMTP Wed, 22 Apr 2020 02:54:56 +0700
EHLO teacher.nt106
250-mail.teacher.nt106 Hello teacher.nt106 [127.0.0.1], please
250-ETRN
250-AUTH LOGIN PLAIN
250-8BITMIME
250-ENHANCEDSTATUSCODES
250-STARTTLS
250 SIZE
AUTH LOGIN
334 UXN1cm5hbWU6
bGFuZG9AdGUhY2hlci5udDEwNg==
334 UGFzc3ducml0MTA2
235 2.7.0 Authentication successful
MAIL FROM:<lando@teacher.nt106>
250 2.1.0 Sender OK
RCPT TO:<hiendo@teacher.nt106>
250 2.1.5 Recipient OK
d
500 5.0.0 Unrecognized command
DATA
354 Enter mail, end with <CRLF>.<CRLF>
To: hiendo@teacher.nt106
From: lando@teacher.nt106
Subject: Test mail From Telnet v2

Hello,
This is a test mail from Telnet
Goodbye.
.
250 2.6.0 Ok, message saved
QUIT
221 2.0.0 See ya in cyberspace
```

Gửi mail với Telnet



The screenshot shows the Microsoft Outlook interface. On the left is a sidebar with a 'Favorites' section containing 'Inbox', 'Sent Items', 'Deleted Items', 'Junk E-mail', 'Outbox', 'RSS Feeds', and 'Search Folders'. Below this is a section for 'hiendo@teacher.nt106' with an 'Inbox' folder highlighted and a count of '1'. The main pane shows a list of emails. The top email is from 'lando@teacher.nt106' with the subject 'Test mail From Telnet v2', received at '3:00 AM', and the body text 'Hello, This is a test mail from'. A magnifying glass icon is positioned over the 'All' filter button. Other emails in the list include 'Microsoft Outlook' (1:39 AM), 'lando@teacher.nt106' (1:36 AM), and 'MDaemon at mail.tea...' (12:57 AM).

Search Current Mailbox Current Mailbox ▼

All Unread By Date ▼ ↑

▼ Today

lando@teacher.nt106
Test mail From Telnet v2 3:00 AM
Hello, This is a test mail from

Microsoft Outlook
Microsoft Outlook Test Message 1:39 AM
This is an email message sent

lando@teacher.nt106
Test mail From Telnet 1:36 AM

MDaemon at mail.tea...
Welcome to the MDaemon em... 12:57 AM
Dear Hien Do,

Test mail From Telnet v2



lando@teacher.nt106
To hiendo@teacher.nt106

Hello,
This is a test mail from Telnet
Goodbye.

SMTP và mã hóa

- SMTP ports:
 - **Port 25** – port không mã hóa
 - **Port 465, 587** – SSL/TLS port, cũng có thể được gọi là SMTPS

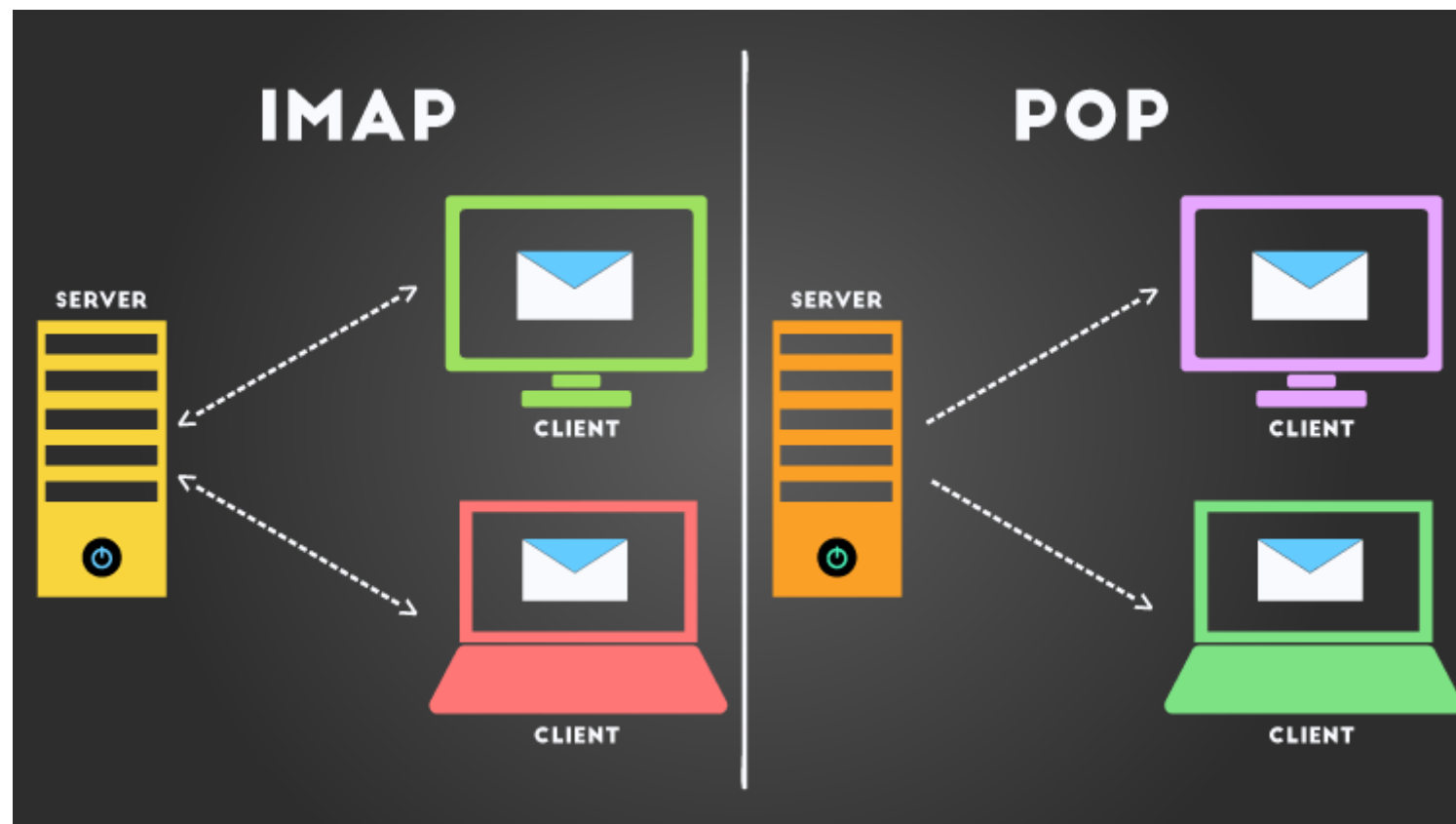
Default Ports:	Server:	Authentication:	Port:
SMTP Server (Outgoing Messages)	Non-Encrypted	AUTH	25 (or 587)
	Secure (TLS)	StartTLS	587
	Secure (SSL)	SSL	465

Nội dung chi tiết

- Giới thiệu về email và nguyên tắc hoạt động
- Giao thức SMTP
- **Giao thức POP3 và IMAP**
- Lập trình Gửi Mail
- Lập trình Nhận Mail

POP3 và IMAP

Giao thức cơ bản để nhận email



POP - Post Office Protocol

- Năm 1984, do máy tính hạn chế tốc độ, băng thông → tạo ra POP
- Năm 1985, POP2 ra đời
- POP3: chuẩn phổ biến nhất, có hỗ trợ chức năng ***leave mail on server***
- POP tải các bản copy của email để đọc khi offline, sau đó xóa những email trên remote server

POP

- Kết nối đến server
- Nhận toàn bộ mail
- Lưu cục bộ như mail mới
- Xóa mail trong server
- Ngắt kết nối

POP3

- Mọi ISP và nhiều công ty cung cấp dịch vụ Web hosting có hỗ trợ POP3 server
- Được định nghĩa trong RFC 1939, hoạt động trên TCP port 110

POP3

- POP3 cũng là giao thức dựa trên dòng lệnh
- Mỗi dòng lệnh kết thúc bằng ký tự linefeed (enter)
- Khi server hoạt động bình thường, mỗi dòng sẽ bắt đầu với +OK.
Nếu lỗi xảy ra, dòng sẽ bắt đầu với -ERR

POP3

- Để truy xuất được mailbox, client phải chứng thực chính nó với username, password

Client gửi:

USER <username><enter>

Server trả lời:

+OK <welcome><enter>

Client gửi mật khẩu:

PASSWORD <password><enter>

POP3

- Để xem trạng thái mailbox:
 - Client gửi:
STAT <enter>
 - Server gửi:
+OK <số thứ tự> <dung lượng><enter>
- Lấy email chỉ định:
 - Client gửi:
RETR <số thứ tự>
(trong đó <số thứ tự> thuộc danh sách đã liệt kê trong lệnh STAT)
 - Server gửi:
+OK <some message>
<enter> <mail body> <enter>.<enter>

POP3

- Để xóa email:
 - Client gửi:
DELE <số thứ tự>
 - Server gửi:
+OK <some message> <enter>
- Tại thời điểm này, email vẫn chưa xóa mà chỉ ở tình trạng bị đánh dấu xóa
- Để hủy đánh dấu xóa:
Client gửi: RSET
- Để kết thúc phiên làm việc, đóng kết nối TCP:
Client dùng lệnh: QUIT <enter>

IMAP - Internet Message Access Protocol

- Ra đời năm 1986
- Ý tưởng: cho phép người dùng xem email trên nhiều thiết bị
- Giữ email trên server cho đến khi người dùng xóa → bị hạn chế dung lượng hộp thư

IMAP

- Email lưu trữ trong IMAP server có thể được đánh dấu như đã trả lời, đính flag, xóa, xem,...
- Các flag giúp một tài khoản IMAP được dùng trên nhiều client. Nếu 1 tài khoản POP3 truy cập trên nhiều client sẽ khó theo dõi tình trạng
- Tương tự POP3 nhưng phức tạp và có cú pháp linh hoạt hơn

IMAP

- Kết nối đến server
- Lấy nội dung được yêu cầu từ người dùng và lưu đệm cục bộ, như danh sách mail mới, nội dung của những email được chọn lựa
- Xử lý các biên tập từ người dùng, như đánh dấu email là mail đã đọc hay xóa...
- Ngắt kết nối

IMAP

- Chạy trên port 143 và được định nghĩa trong RFC 1730
- SMTP và POP3 là các chuẩn của truyền thông email

IMAP

Tiến trình làm việc tương tự với POP:

- Chứng thực:
 - Client gửi login <username> <password>
 - Nếu đúng thì server trả lời: OK
 - LOGIN completed
- Lấy thông tin về mailbox:
 - clien gửi select inbox.
 - Server trả lời: * <số lượng mail>
 - EXISTS
- Lấy một email: clien gửi fetch <số thứ tự>
- Server trả lời với nội dung email theo chuẩn RFC 822 và thông báo OK FETCH completed
- Xóa một email: clien gửi store <số thứ tự> +flags \deleted

Lựa chọn POP hay IMAP?

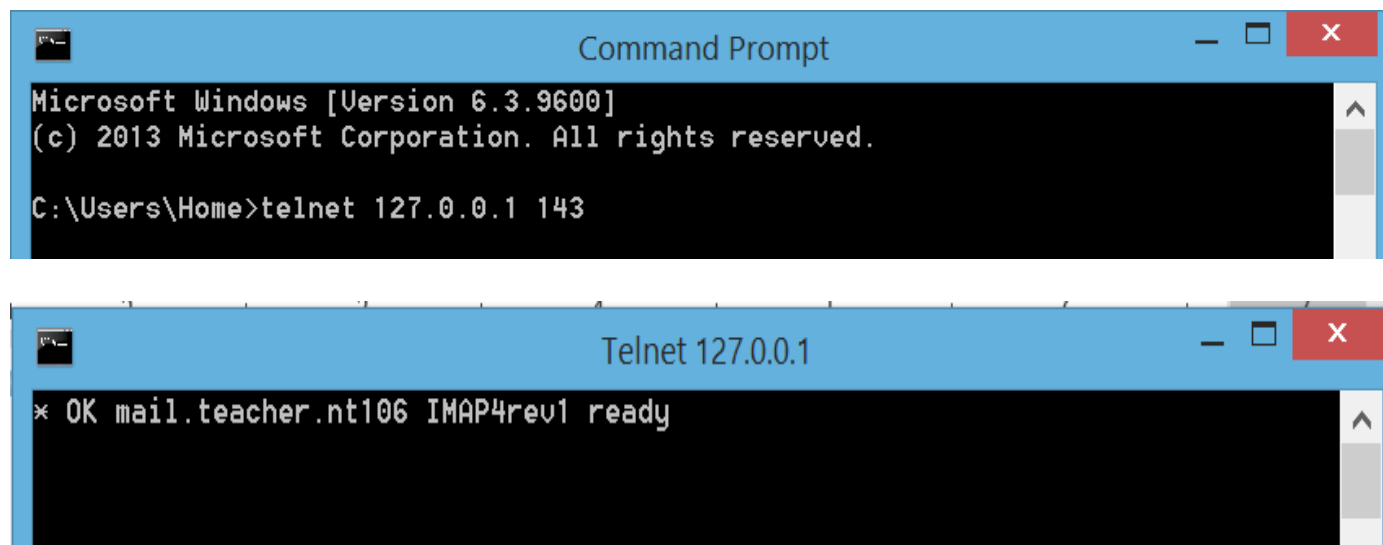


Lựa chọn POP hay IMAP?

- Chọn **POP** nếu:
 - Muốn truy cập mail chỉ từ một thiết bị
 - Cần truy cập email thường xuyên dù có kết nối Internet hay không
 - Không gian lưu trữ trên server hạn chế
- Chọn **IMAP** nếu:
 - Muốn truy cập email từ nhiều thiết bị khác nhau
 - Có kết nối Internet thường xuyên và tin cậy
 - Muốn xem nhanh các email mới hoặc những email trên server
 - Không gian lưu trữ cục bộ hạn chế
 - Lo lắng về vấn đề dự phòng dữ liệu

Nhận mail

- Mở cmd
- Gõ lệnh: `telnet <server name> 110 hoặc 143`



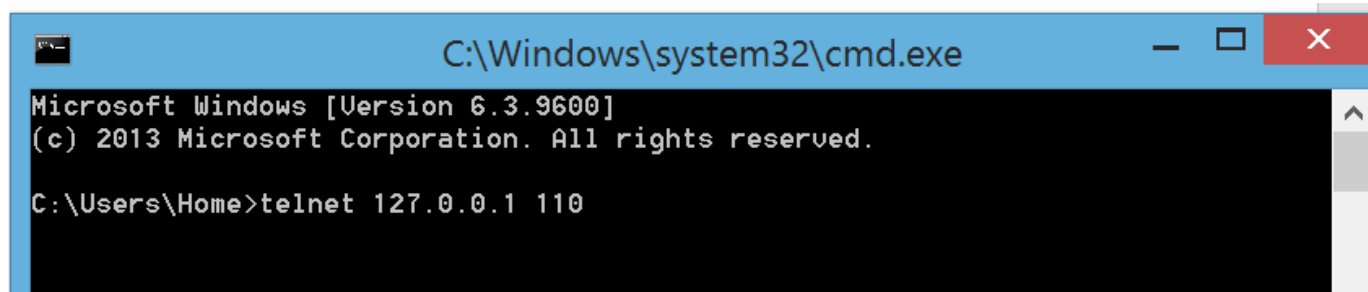
The image shows two screenshots of Windows command prompts. The top screenshot is titled 'Command Prompt' and shows the command 'telnet 127.0.0.1 143' being entered. The bottom screenshot is titled 'Telnet 127.0.0.1' and shows the response 'x OK mail.teacher.nt106 IMAP4rev1 ready'.

```
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\Home>telnet 127.0.0.1 143

Telnet 127.0.0.1
x OK mail.teacher.nt106 IMAP4rev1 ready
```

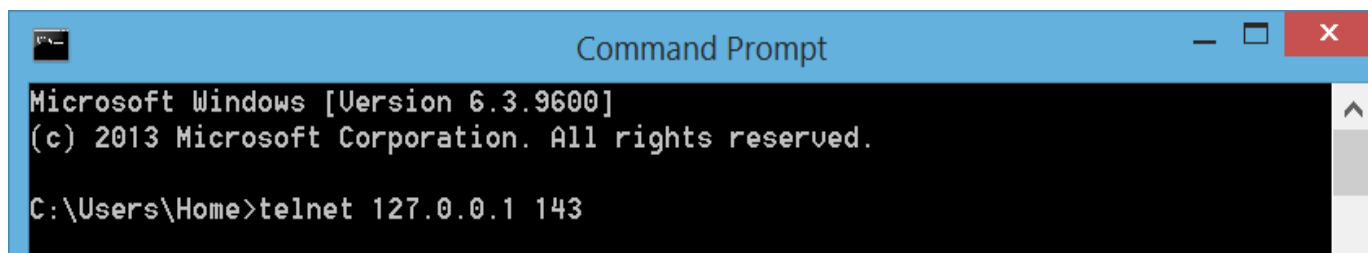

Nhận mail

- Mở cmd
- Gõ lệnh: `telnet <server name> 110` hoặc `143`



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Home>telnet 127.0.0.1 110
```



```
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Home>telnet 127.0.0.1 143
```

Nhận mail (POP3)

```
Telnet 127.0.0.1
+OK mail.teacher.nt106 POP3 ready
USER hiendo@teacher.nt106
+OK hiendo@teacher.nt106... User ok
PASS Uit106
+OK hiendo@teacher.nt106's mailbox has 6 total messages (10262 octets)
LIST
+OK 6 10262
1 4850
2 953
3 1675
4 905
5 870
6 1009
.
r
-ERR unknown POP3 command!
RETR 2
+OK 953 octets
X-MDAU-Result: clean
X-MDAU-Processed: mail.teacher.nt106, Wed, 22 Apr 2020 01:36:20 +0700
Return-path: <lando@teacher.nt106>
Authentication-Results: mail.teacher.nt106;
    auth=pass (login) smtp.auth=lando@teacher.nt106
Received: from teacher.nt106 by mail.teacher.nt106 with ESMTPA id md500000000003.
msg;
    Wed, 22 Apr 2020 01:36:19 +0700
X-Spam-Processed: mail.teacher.nt106, Wed, 22 Apr 2020 01:36:19 +0700
    (not processed: message from trusted or authenticated source)
X-MDArrival-Date: Wed, 22 Apr 2020 01:36:19 +0700
X-Authenticated-Sender: lando@teacher.nt106
X-Rcpt-To: hiendo@teacher.nt106
```

Nhận mail (IMAP)

```
Telnet 127.0.0.1
* OK mail.teacher.nt106 IMAP4rev1 ready
a1 LOGIN lando@teacher.nt106 Uit106
a1 OK LOGIN completed
a2 LIST "" "*"
* LIST () "/" "INBOX"
* LIST (\Trash) "/" "Deleted Items"
* LIST (\Drafts) "/" "Drafts"
* LIST () "/" "Junk Email"
* LIST (\Sent) "/" "Sent Items"
a2 OK LIST completed
a3 EXAMINE INBOX
* FLAGS (\Seen \Answered \Flagged \Deleted \Draft \Recent $Forwarded $MDNSent)
* 6 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1587492572] UIDs valid
* OK [UIDNEXT 7] Predicted next UID
* OK [PERMANENTFLAGS ()] No permanent flags permitted
a3 OK [READ-ONLY] EXAMINE completed
a4 FETCH 6 BODY[]
* 6 FETCH (BODY[] {5175})
Return-Path: <>
From: "MDaemon at mail.teacher.nt106" <postmaster@teacher.nt106>
To: lando@teacher.nt106
Subject: WARNING: Message delivery failed
Date: Wed, 22 Apr 2020 01:19:36 +0700
```

Nội dung chi tiết

- Giới thiệu về email và nguyên tắc hoạt động
- Giao thức SMTP
- Giao thức POP3 và IMAP
- **Lập trình Gửi Mail**
- Lập trình Nhận Mail

Lập trình gửi mail bằng TcpClient

- Khai báo thư viện:
`using System.Threading;`
`using System.Net;`
`using System.Net.Sockets;`
`using System.Text;`
`using System.IO;`

Lập trình gửi mail bằng TCPClient

- Khai báo đối tượng thuộc lớp TcpClient để quản lý kết nối với server:

```
TcpClient tcpclient;  
tcpclient = new TcpClient();  
tcpclient.Connect("127.0.0.1", 25);
```

Lập trình gửi mail bằng TCPClient

- Sử dụng StreamReader và StreamWriter để đọc và ghi dữ liệu

```
StreamReader sr = new StreamReader(tcpclient.GetStream());  
StreamWriter sw = new StreamWriter(tcpclient.GetStream());
```

Lập trình gửi mail bằng TCPClient

- Gửi lời chào cho server:

```
data = "EHLO teacher.nt106";  
sw.WriteLine(data);  
sw.Flush();
```

- Chọn xác thực:

```
data = "AUTH LOGIN";  
sw.WriteLine(data);  
sw.Flush();
```


Lập trình gửi mail bằng TCPClient

- Hàm chuyển chuỗi sang dạng base64:

```
static public string EncodeTo64(string toEncode)
{
    byte[] toEncodeAsBytes =
        System.Text.ASCIIEncoding.ASCII.GetBytes(toEncode);
    string returnValue = System.Convert.ToBase64String(toEncodeAsBytes);
    return returnValue;
}
```

Lập trình gửi mail bằng TCPClient

Đối với SMTP, chỉ có thể dùng ASCII.GetBytes vì SMTP là giao thức gửi dựa trên văn bản, dữ liệu nhị phân không được chấp nhận

Lập trình gửi mail bằng TCPClient

- Gửi thông tin username đã chuyển sang base64:

```
string username = EncodeTo64(mailfrom);  
sw.WriteLine(username);  
sw.Flush();
```

- Gửi thông tin password đã chuyển sang base64 :

```
sw.WriteLine(EncodeTo64(password));  
sw.Flush();
```

Lập trình gửi mail bằng TCPClient

- Gửi địa chỉ sender cho server:

```
data = "MAIL FROM:<" + mailfrom + ">";  
sw.WriteLine(data);  
sw.Flush();
```

- Gửi địa chỉ receiver cho server:

```
data = "RCPT TO:<" + mailto + ">";  
sw.WriteLine(data);  
sw.Flush();
```

Lập trình gửi mail bằng TCPClient

- Bắt đầu nội dung thư:

```
data = "DATA";  
sw.WriteLine(data);  
sw.Flush();
```

- Thiết lập subject:

```
data = "Subject: " + Subject;  
sw.WriteLine(data);  
sw.Flush();
```

Lập trình gửi mail bằng TCPClient

- Ngắt kết nối:

```
data = "QUIT";  
sw.WriteLine(data);  
sw.Flush();
```

Lập trình gửi mail bằng TCPClient

Form1

From: Password:

To:

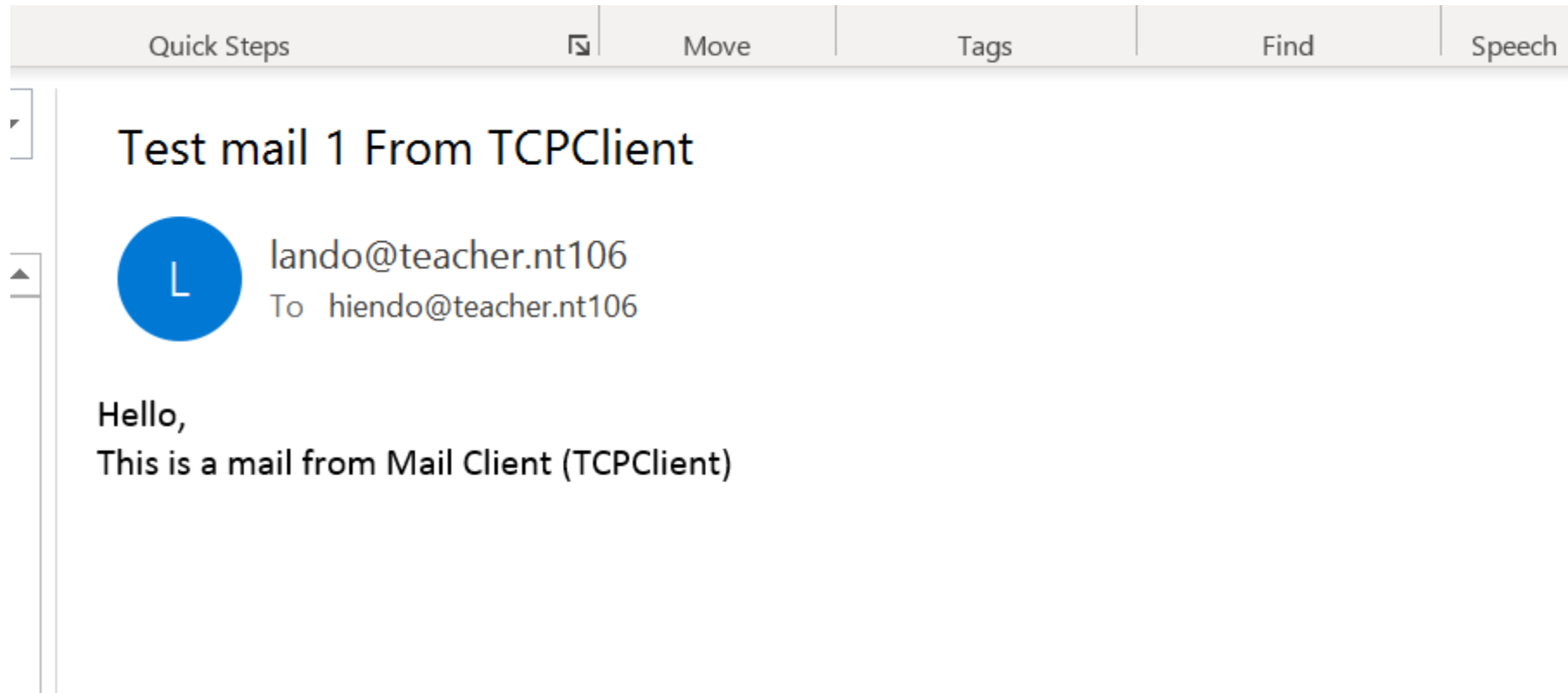
Subject:

Body:

```
S: 220 mail.teacher.nt106 ESMTP Wed, 22 Apr 2020 04:46:27 +0700
S: 250-mail.teacher.nt106 Hello teacher.nt106 [127.0.0.1], pleased to meet you
S: 250-ETRN
S: 250-AUTH LOGIN PLAIN
S: 250-8BITMIME
S: 250-ENHANCEDSTATUSCODES
S: 250-STARTTLS
S: 250 SIZE
S: 334 VXNlcm5hbWU6
S: 334 UGFzc3dvcmQ6
S: 235 2.7.0 Authentication successful
S: 250 2.1.0 Sender OK
S: 250 2.1.5 Recipient OK
S: 354 Enter mail, end with <CRLF>.<CRLF>
S: 250 2.6.0 Ok, message saved

C: EHLO teacher.nt106
C: AUTH LOGIN
C: bGFuZG9AdGVhY2hlcj5udDEwNg==
C: VVl0MTA2
C: MAIL FROM:<lando@teacher.nt106>
C: RCPT TO:<hiendo@teacher.nt106>
C: DATA
C: Subject: Test mail 1 From TCPClient
C: From: lando@teacher.nt106
C: To: hiendo@teacher.nt106
C:
C: Hello, This is a mail from Mail Client (TCPClient)
C: .
C: QUIT
```

Lập trình gửi mail bằng TCPClient



Lập trình gửi mail bằng SmtplibClient

Tham khảo thêm tại:

<https://docs.microsoft.com/en-us/dotnet/api/system.net.mail.smtpclient?view=netframework-4.8>

Lập trình gửi mail bằng Smtplib

```
private void sendEmail_click(object sender, EventArgs e)
{
    using (SmtplibClient smtpClient = new SmtplibClient("127.0.0.1"))
    {
        string mailfrom = txtMailFrom.Text.ToString().Trim();
        string mailto = txtMailTo.Text.ToString().Trim();
        string password = txtPass.Text.ToString().Trim();
        var basicCredential = new NetworkCredential(mailfrom, password);
        using (MailMessage message = new MailMessage())
        {
            MailAddress fromAddress = new MailAddress(mailfrom);
            smtpClient.UseDefaultCredentials = false;
            smtpClient.Credentials = basicCredential;
```

Lập trình gửi mail bằng Smtplib

```
message.From = fromAddress;
message.Subject = txtSubject.Text.ToString().Trim() + " From
SMTPClient";
// Set IsBodyHtml to true means you can send HTML email.
message.IsBodyHtml = true;
message.Body = rtbBody.Text.ToString();
message.To.Add(mailto);
try
{
    smtpClient.Send(message);
}
catch (Exception ex)
{
    MessageBox.Show(ex.ToString());
}
}
}
```

Lập trình gửi mail bằng SmtplibClient

Test mail 2 From SMTPClient



lando@teacher.nt106

To hiendo@teacher.nt106

Hello, This is a mail from Mail Client

Lập trình gửi mail bằng MailKit

Remarks

The `SmtpClient` class is used to send email to an SMTP server for delivery. The SMTP protocol is defined in RFC 2821, which is available at <https://www.ietf.org>.

❗ Important

We don't recommend that you use the `SmtpClient` class for new development because `SmtpClient` doesn't support many modern protocols. Use [MailKit](#) or other libraries instead. For more information, see [SmtpClient shouldn't be used](#) on GitHub.

<https://github.com/jstedfast/MailKit>

Nội dung chi tiết

- Giới thiệu về email và nguyên tắc hoạt động
- Giao thức SMTP
- Giao thức POP3 và IMAP
- Lập trình Gửi Mail
- **Lập trình Nhận Mail**

Lập trình nhận mail POP3 bằng TCPClient

- Khai báo thư viện:

```
using System.Threading;  
using System.Net;  
using System.Net.Sockets;  
using System.Text;  
using System.IO;
```

Lập trình nhận mail POP3 bằng TCPClient

- Khai báo đối tượng TcpClient và các Stream để đọc, ghi:

```
TcpClient popclient;
```

```
StreamReader sr;
```

```
StreamWriter sw;
```


Lập trình nhận mail POP3 bằng TCPClient

- Xử lý sự kiện Login:

```
private void btnLogin_Click(object sender, EventArgs e)
{
    IPAddress server;
    IPAddress.TryParse(txtServer.Text, out server);
    int port;
    Int32.TryParse(txtPort.Text.Trim(), out port);
    IPEndPoint iep = new IPEndPoint(server, port);
    popclient = new TcpClient();
    popclient.Connect(iep);
    sr = new StreamReader(popclient.GetStream());
    sw = new StreamWriter(popclient.GetStream());
    sr.ReadLine();
}
```

Lập trình nhận mail POP3 bằng TCPClient

- Gửi lệnh USER và PASS để đăng nhập:

```
string data = "";  
data = "USER " + txtUser.Text;  
sw.WriteLine(data);  
sw.Flush();  
sr.ReadLine();
```

```
data = "PASS " + txtPass.Text;  
sw.WriteLine(data);  
sw.Flush();  
sr.ReadLine();
```

Lập trình nhận mail POP3 bằng TCPClient

- Gửi lệnh LIST để lấy danh sách mail, và đổ vào ListBox:

```
data = "LIST";
sw.WriteLine(data);
sw.Flush();

listBox1.Items.Clear();
string s = sr.ReadLine();
while ((s = sr.ReadLine()) != ".")
{
    listBox1.Items.Add(s);
}
}
```

Lập trình nhận mail POP3 bằng TCPClient

- Hàm xử lý chọn email từ danh sách để xem nội dung

```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    int i = listBox1.SelectedIndex + 1;
    //Lay buc thu ve va tien hanh phan tich
    string data = "RETR " + i.ToString();
    sw.WriteLine(data);
    sw.Flush();
    string s;
```

Lập trình nhận mail POP3 bằng TCPClient

- Hàm xử lý chọn email từ danh sách để xem nội dung (tt)

```
//Lay phan header
```

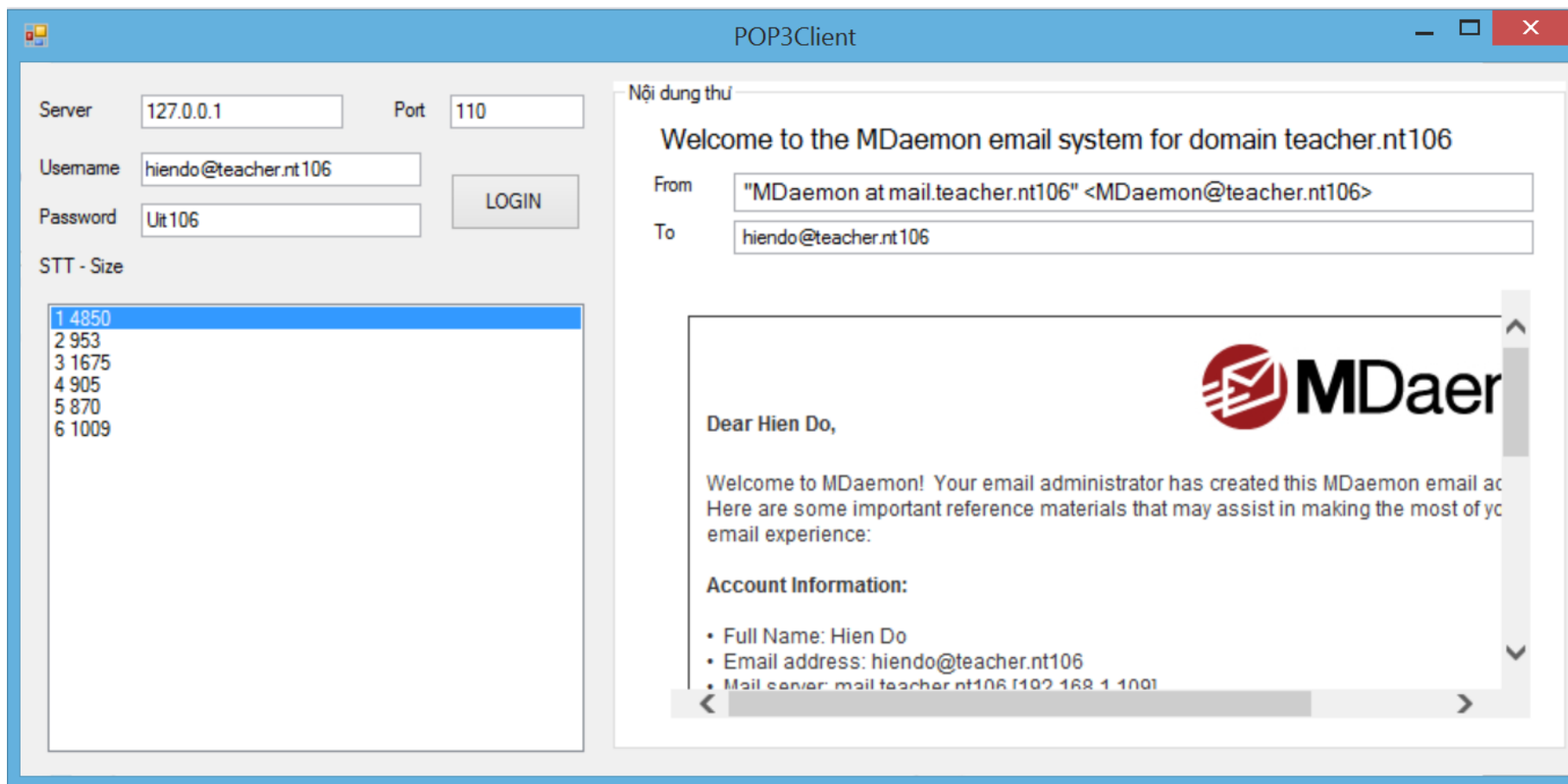
```
while ((s = sr.ReadLine().Trim()) != null)
{
    if (s.Length == 0) break;
    if (s.ToUpper().StartsWith("FROM"))
        txtFrom.Text = s.Substring(5, s.Length - 5);
    if (s.ToUpper().StartsWith("TO"))
        txtTo.Text = s.Substring(3, s.Length - 3);
    if (s.ToUpper().StartsWith("SUBJECT"))
        txtSubject.Text = s.Substring(8, s.Length - 8);
}
```

Lập trình nhận mail POP3 bằng TCPClient

- Hàm xử lý chọn email từ danh sách để xem nội dung (tt)

```
//Lay phan body
string html = "";
while (!sr.EndOfStream)
{
    s = sr.ReadLine().Trim();
    if (s.Equals("."))
    {
        break;
    }
    html += s + "\r\n";
}
//Hien thi Email (HTML) bang WebBrowser
wcBody.DocumentText = html;
Application.DoEvents();
}
```

Lập trình nhận mail POP3 bằng TCPCClient



Lập trình nhận mail IMAP với Mailkit

```
using System;

using MailKit.Net.Imap;
using MailKit.Search;
using MailKit;
using MimeKit;

namespace TestClient {
    class Program
    {
        public static void Main (string[] args)
        {
            using (var client = new ImapClient ()) {
                client.Connect ("imap.friends.com", 993, true);

                client.Authenticate ("joey", "password");

                // The Inbox folder is always available on all IMAP servers...
                var inbox = client.Inbox;
                inbox.Open (FolderAccess.ReadOnly);
            }
        }
    }
}
```

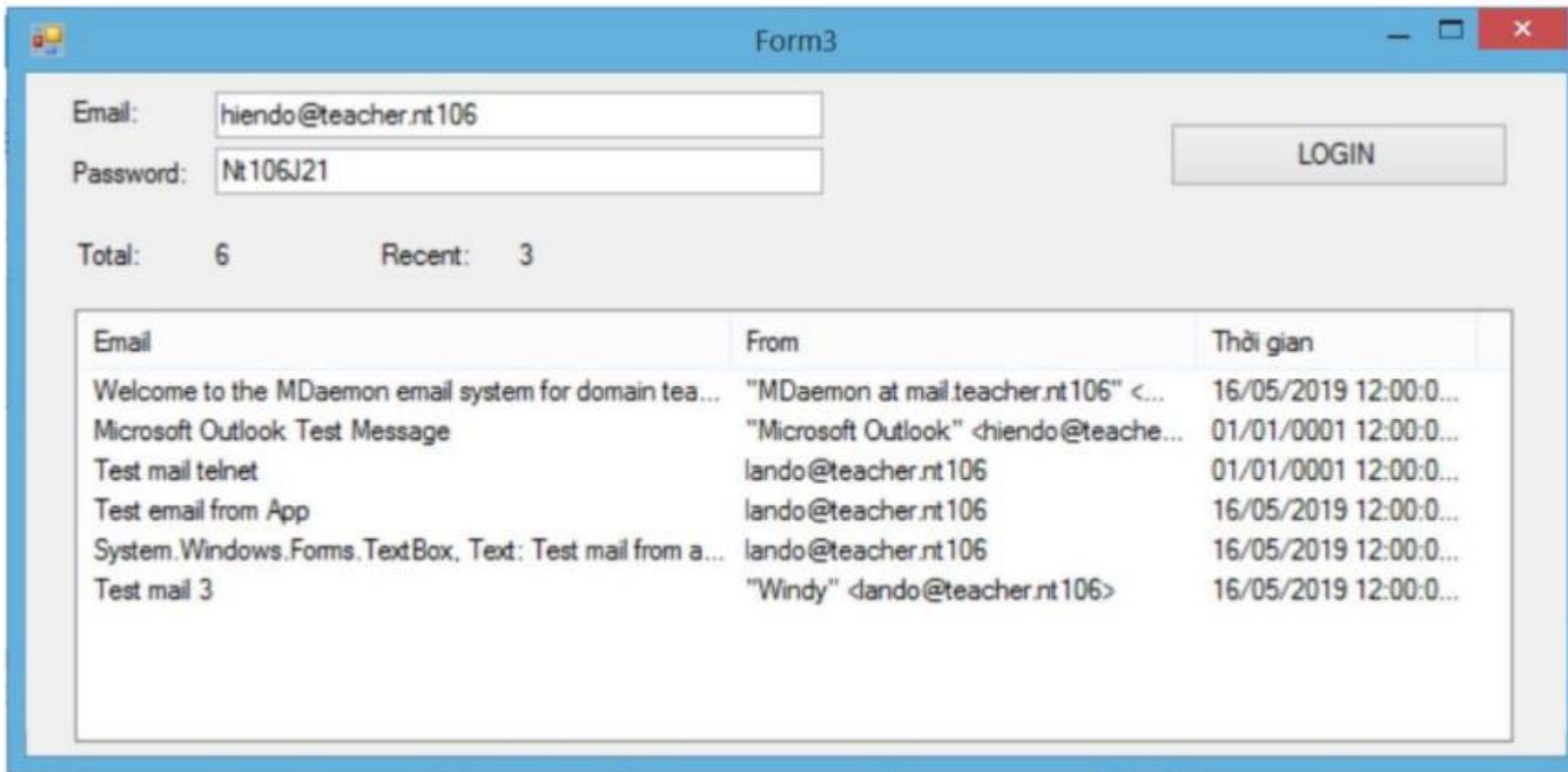

Lập trình nhận mail IMAP với Mailkit

```
        Console.WriteLine ("Total messages: {0}", inbox.Count);
        Console.WriteLine ("Recent messages: {0}", inbox.Recent);

        for (int i = 0; i < inbox.Count; i++) {
            var message = inbox.GetMessage (i);
            Console.WriteLine ("Subject: {0}", message.Subject);
        }

        client.Disconnect (true);
    }
}
}
```

Lập trình nhận mail IMAP với Mailkit



The screenshot shows a Windows application window titled "Form3". It contains a login form with the following fields and buttons:

- Email:
- Password:
- LOGIN button

Below the login fields, it displays statistics:

Total: 6 Recent: 3

A table of received emails is shown below:

Email	From	Thời gian
Welcome to the MDaemon email system for domain tea...	"MDaemon at mail.teacher.nt106" <...>	16/05/2019 12:00:0...
Microsoft Outlook Test Message	"Microsoft Outlook" <hiendo@teache...	01/01/0001 12:00:0...
Test mail telnet	lando@teacher.nt106	01/01/0001 12:00:0...
Test email from App	lando@teacher.nt106	16/05/2019 12:00:0...
System.Windows.Forms.TextBox, Text: Test mail from a...	lando@teacher.nt106	16/05/2019 12:00:0...
Test mail 3	"Windy" <lando@teacher.nt106>	16/05/2019 12:00:0...